CLAIMS:

What is claimed is:

1. A method of autonomically reorganizing code of a computer program, comprising the steps of:

monitoring branch count per instruction statistics, wherein the branch count per instruction statistics are generated from the results of a set of hardware counters that count branches taken per instruction of the computer program;

determining whether a block of code is to be reorganized, wherein the block of code comprises a set of instructions;

in response to the step of determining, locally reorganizing the block of code such that fewer branches are taken.

2. The method of claim 1, wherein the step of determining whether a block of code is to be reorganized is based on the branch count per instruction statistics.

3. The method of claim 1, wherein prior to the step of reorganizing the block of code, execution of the computer program is halted.

4. The method of claim 1, wherein reorganization of the block of code results in instructions of the block of code being more contiguous.

5. The method of claim 1, wherein reorganizing the block of code is performed locally by modifying an if/then/else clause condition.

6. The method of claim 1, wherein reorganization of the block of code is performed locally by switching a then/else statement of an if/then/else clause of a branch instruction of the block of code.

7. A computer system for autonomically reorganizing code of a computer program, comprising:

a set of hardware counters associated with a set of branch instructions of a computer program, wherein the hardware counters are used to generate branch count per instruction statistics;

a block of code including at least one branch instruction of the set of branch instructions;

wherein the block of code is locally reorganized; and

wherein the branch count per instruction statistics are used to determine whether to autonomically reorganize a block of code.

8. The system of claim 7, wherein the block of code is locally reorganized by modifying an if/then/else clause condition.

9. The system of claim 7, wherein the block of code is locally reorganized by switching a then/else statement of

an if/then/else clause of an instruction of the block of code.

10. The system of claim 7, wherein execution of the computer program is halted while the block of code is locally reorganized.

11. The system of claim 7, wherein local reorganization of the block of code results in fewer branches being taken during execution of the program.

12. A computer program product in a computer readable medium for autonomically reorganizing code of a computer program, comprising:

first instructions for monitoring branch count per instruction statistics, wherein the branch count per instruction statistics are generated from the results of a set of hardware counters that count branches taken per instruction of the computer program;

second instructions for determining whether a block of code is to be reorganized, wherein the block of code comprises a set of instructions;

third instructions for, in response to the step of determining, locally reorganizing the block of code such that fewer branches are taken.

13. The computer program product of claim 12, wherein determining whether a block of code is to be reorganized is based on the branch count per instruction statistics.

14. The computer program product of claim 12, wherein prior to reorganizing the block of code, execution of the computer program is halted.

15. The computer program product of claim 12, wherein reorganizing the block of code results in instructions of the block of code being more contiguous.

16. The computer program product of claim 12, wherein reorganizing the block of code is performed locally by modifying an if/then/else clause condition.

17. The computer program product of claim 12, wherein reorganizing the block of code is performed locally by switching a then/else statement of an if/then/else clause of an instruction of the block of code.